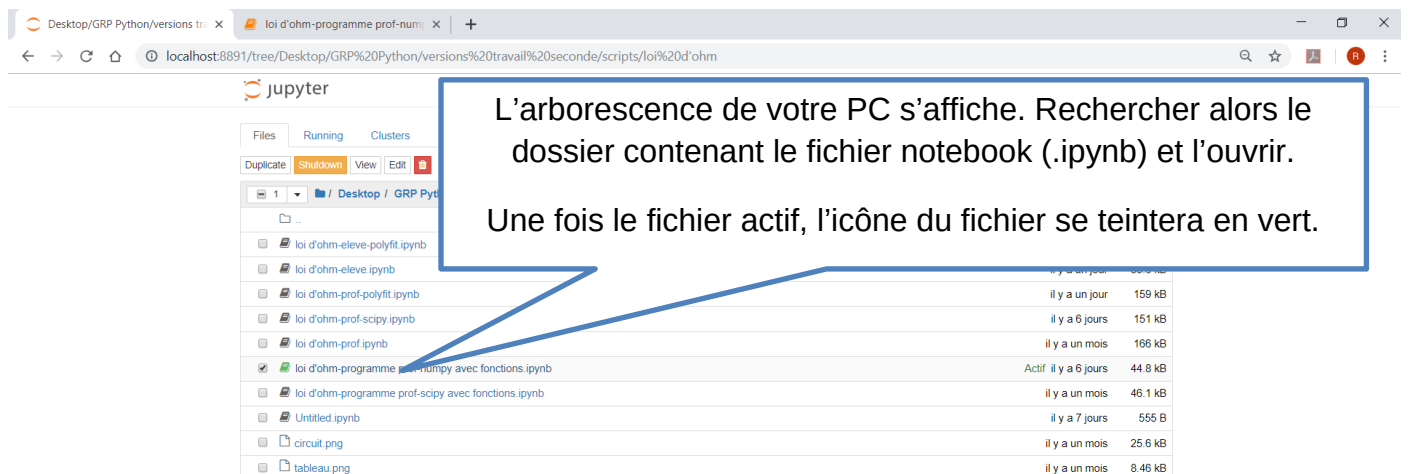
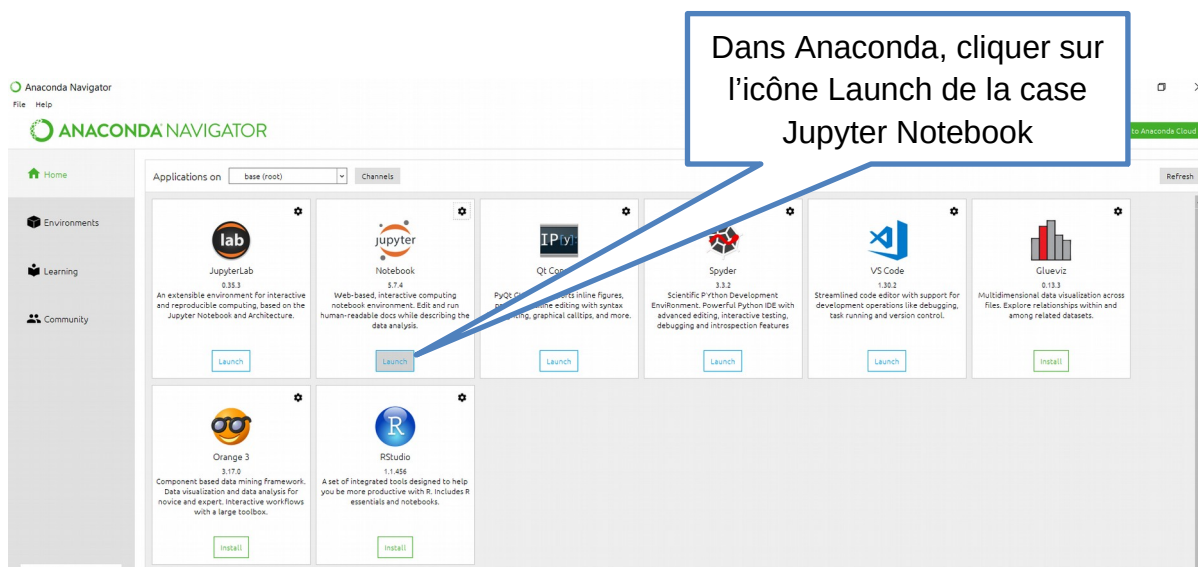


Guide d'utilisation rapide de Notebook Jupyter

Après avoir téléchargé Anaconda Navigator sur votre ordinateur :



Remarque : Dans le cas où votre fichier Notebook se situe dans le dossier Mes documents de votre ENT, ouvrez directement le fichier en cliquant sur son nom.

Le fichier notebook s'affiche dans une nouvelle fenêtre, l'arborescence restant alors accessible dans la fenêtre précédente.



Un fichier notebook se présente comme une succession de cellules qui sont de deux types :

- des cellules « Markdown » permettant d'écrire du texte (titre, contexte de l'activité, consignes élèves...), d'insérer des fichiers image...
- des cellules « Code » permettant d'écrire des lignes de code en langage Python puis de les exécuter. Le mot « Entrée [] » (ou « In [] ») est écrit dans la marge à gauche dans ce type de cellule.

Copie d'écran avec des cellules « Markdown » et une cellule « Code ».

The screenshot shows a Jupyter Notebook interface with four cells. The first cell is a Markdown cell containing the title "# Activité : La loi d'Ohm". The second cell is a Markdown cell containing the command "![tableau.png](attachment:tableau.png)". The third and fourth cells are Code cells, each starting with "Entrée []:". The third code cell contains a function definition for "modelisation(x,y)" that uses numpy and matplotlib to fit a polynomial and print the coefficients. The fourth code cell contains a function definition for "courbemodelisee(x,y,ymodel)" that plots the experimental data points and the fitted linear model on a graph.

Comment utiliser un fichier Notebook ?

Pour utiliser votre fichier Notebook, il faut exécuter les cellules du Notebook dans l'ordre, les unes après les autres.

Pour cela :

1. Sélectionner la cellule, un cadre autour de la cellule apparaît
2. Taper **Ctrl+Entrée** ou cliquer sur la commande «Exécuter»

This screenshot shows the same Jupyter Notebook as above, but with callouts highlighting the execution process. A blue box around the first cell is labeled "1. Sélectionner la cellule, un cadre autour de la cellule apparaît". Another blue box around the "Exécuter" button in the toolbar is labeled "2. Taper Ctrl+Entrée ou cliquer sur la commande «Exécuter»". The notebook interface includes a menu bar (File, Edit, View, Kernel, Widgets, Help), a toolbar with various icons, and a status bar showing "De Confiance" and "Python 3".

Copie d'écran avec les cellules « Markdown » et « Code » précédentes exécutées.

File Edit View Insert Cell Kernel Widgets Help De Confiance Python 3.0

Exécuter Code

Cellule «Markdown» exécutée

Activité : La loi d'Ohm

I(mA)	0	25	50	75	100	125
U(V)	0	1,7	3,4	5,1	6,8	8,5

```

Entrée [1]: 1 import numpy as np
            2 import matplotlib.pyplot as plt

Entrée [2]: 1 # fonction modélisation par une droite d'équation y=ax+b (polynôme d'ordre 1)
            2 def modelisation(x,y):
            3     coef=np.polyfit(x,y,1) # permet de déterminer les coefficients a=coef[0] et b=coef[1] de la droite
            4     ymodel=coef[0]*x+coef[1] # permet de déterminer les valeurs modélisées de y
            5     print ('U= %.1f*I'%coef[0])
            6     return (ymodel)

Entrée [3]: 1 # fonction permettant de tracer le graphique avec les points expérimentaux et la courbe obtenue après modélisation
            2 def courbemodélisee(x,y,ymodel):
            3     fig = plt.figure(figsize=(12,10))
            4     plt.plot(x,y,'r+',label='U=f(I)')
            5     plt.plot(x,ymodel,'b',label='modèle linéaire')
  
```

Cellule «Code» exécutée

- Si la cellule est de type « Markdown » : les textes, les images jointes... entrés dans la cellule apparaissent dans leur forme lisible par l'utilisateur du fichier Notebook. Lors de l'ouverture du fichier, ces cellules ont souvent déjà été exécutées par le créateur du fichier, ce qui permet de faire de belles présentations notamment à destination des élèves.
- Si la cellule est de type « Code » : Lors de l'exécution, une étoile * apparaît entre les crochets puis une fois la cellule exécutée, le « numéro d'ordre d'exécution » s'affiche.

Il est très important d'exécuter les cellules de code dans l'ordre du programme. Les nombres entre crochets peuvent ne pas se suivre un par un notamment si vous exécutez une même cellule plusieurs fois de suite mais l'ordre de ces nombres doit être croissant au fur et à mesure que l'on avance dans le fichier Notebook.

```

Entrée [1]: 1 import numpy as np
            2 import matplotlib.pyplot as plt

Entrée [2]: 1 # fonction modélisation par une droite d'équation y=ax+b (polynôme d'ordre 1)
            2 def modelisation(x,y):
            3     coef=np.polyfit(x,y,1) # permet de déterminer les coefficients a=coef[0] et b=coef[1] de la droite
            4     ymodel=coef[0]*x+coef[1] # permet de déterminer les valeurs modélisées de y
            5     print ('U= %.1f*I'%coef[0])
            6     return (ymodel)

Entrée [3]: 1 # fonction permettant de tracer le graphique avec les points expérimentaux et la courbe obtenue après modélisation
            2 def courbemodélisee(x,y,ymodel):
            3     fig = plt.figure(figsize=(12,10))
            4     plt.plot(x,y,'r+',label='U=f(I)')
            5     plt.plot(x,ymodel,'b',label='modèle linéaire')
  
```

Une fois exécutée, le résultat de l'exécution (sortie/output) s'affiche dans la cellule sous le code (cela peut être un message d'erreur si votre code est erroné). Il peut ne rien s'afficher si le code ne le demande pas.

Il est possible d'exécuter une même cellule plusieurs fois de suite lorsque vous voulez modifier et tester le code qu'elle contient. Dans ce cas, il n'est pas nécessaire de réexécuter les cellules précédentes, ni de supprimer la sortie précédente de la cellule concernée car elle sera remplacée automatiquement lors de la nouvelle exécution de la cellule.

```

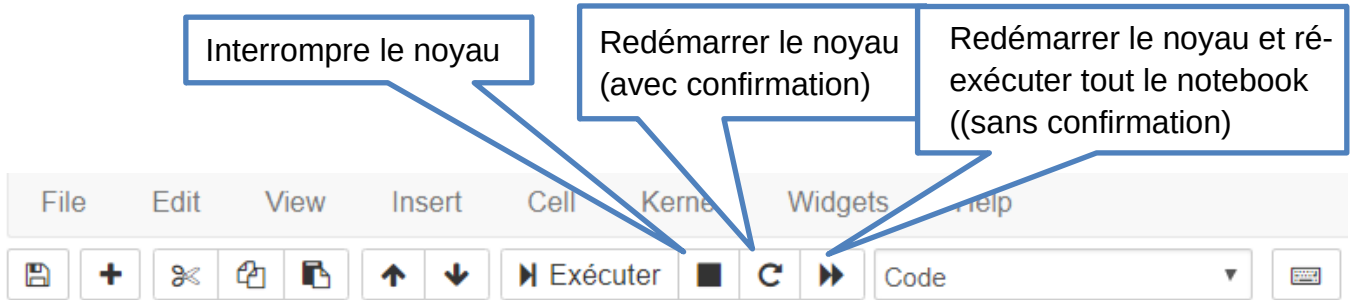
Entrée [12]: 1 I=np.array([0,25e-3,50e-3,75e-3,100e-3,125e-3])
            2 U=np.array([0,1.7,3.4,5.1,6.8,8.5])
            3 Umodel=modelisation(I,U)
            4 courbemodélisee(I,U,Umodel)
  
```

U= 68.0xI

Caractéristique Intensité-Tension d'un dipôle ohmique

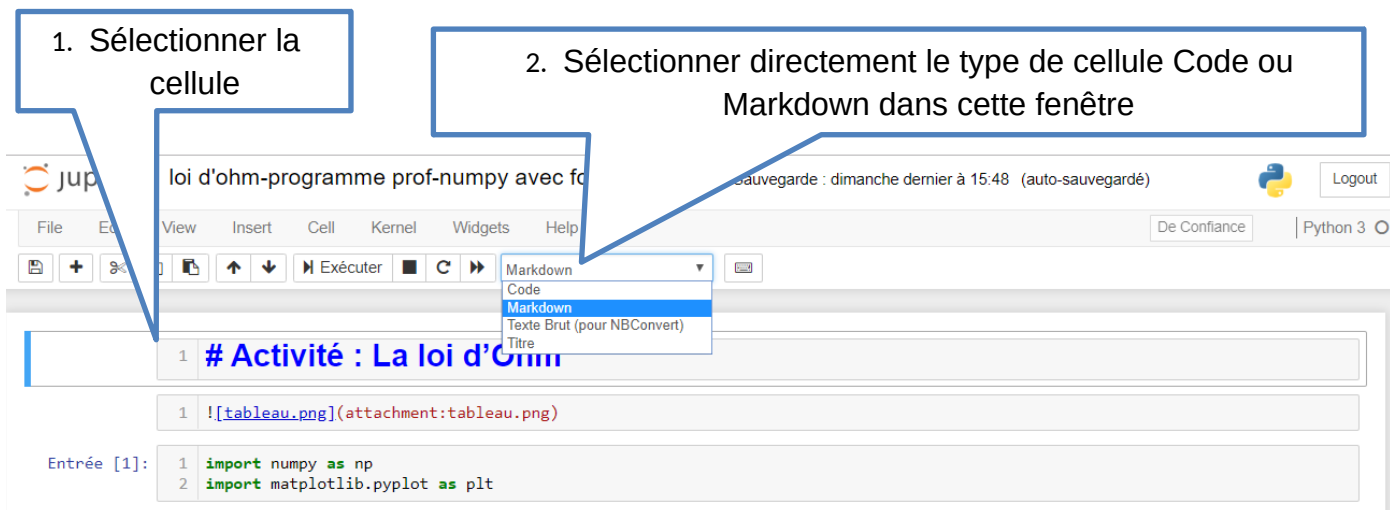
Enfin, il est possible de réexécuter le fichier Notebook depuis la première cellule après une ou plusieurs exécutions.

Cliquer alors sur les commandes « interrompre le noyau » puis « redémarrer le noyau (avec confirmation) » ou « redémarrer le noyau et ré-exécuter tout le notebook (sans confirmation) »

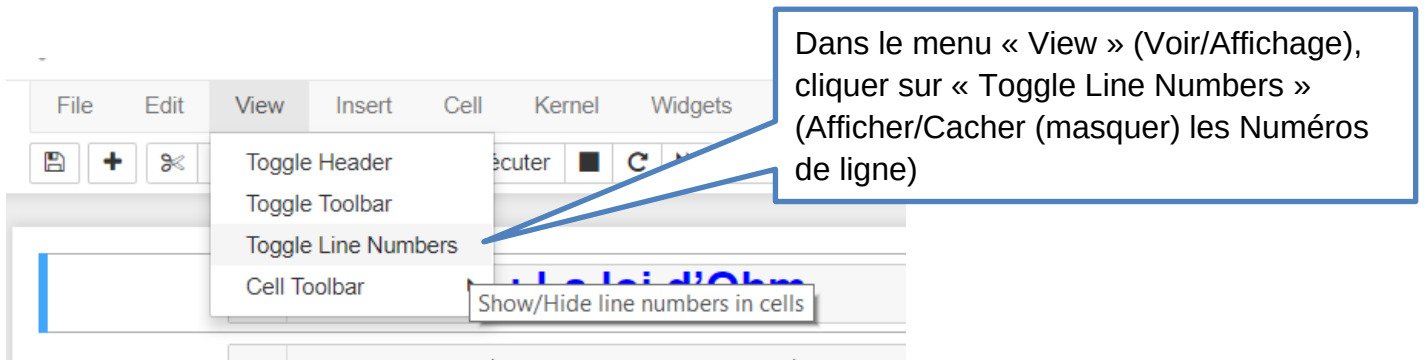


Attention : « Interrompre le noyau » annule l'exécution de toutes les cellules du fichier Notebook (même si les nombres entre crochets ne s'effacent pas). Il faut donc recommencer l'exécution des cellules dès le début du fichier.

Comment définir le type de cellule ? (Markdown ou Code)



Comment numéroté les lignes d'une cellule ?

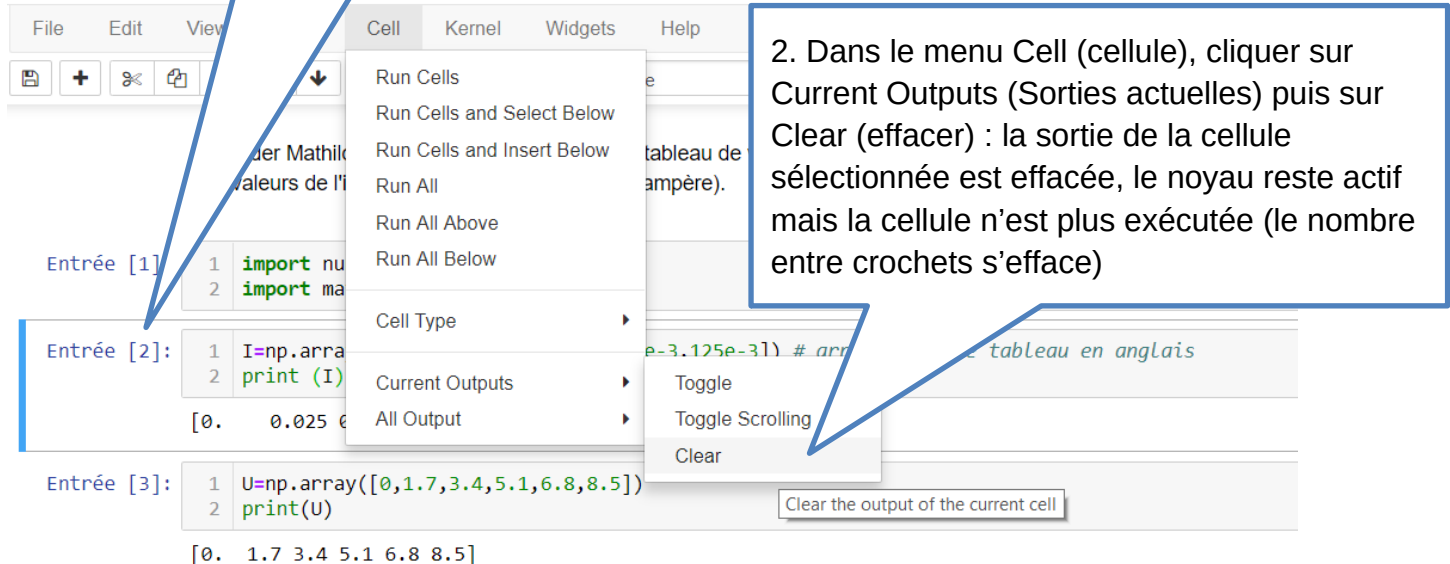


Les numéros de ligne s'affichent pour toutes les cellules du notebook

```
1 # Activité : La loi d'Ohm
1 ![tableau.png](attachment:tableau.png)
Entrée [1]: 1 import numpy as np
2 import matplotlib.pyplot as plt
Entrée [2]: 1 # fonction modélisation par une droite d'équa
2 def modelisation(x,y):
3     coef=np.polyfit(x,y,1) # per
4     ymodel=coef[0]*x+coef[1] # per
5     print ('U= %.1fV'%coef[0])
6     return (ymodel)
```

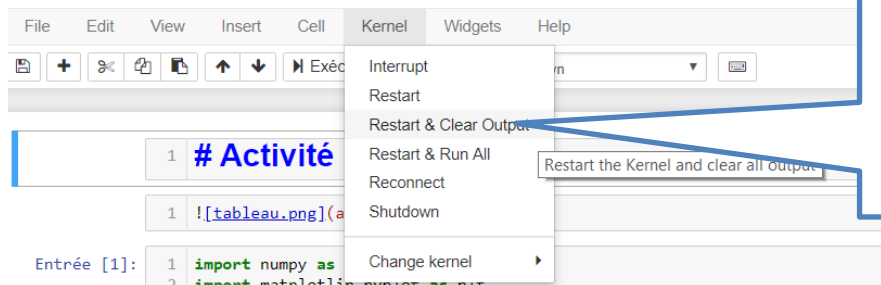
Comment effacer la sortie d'une seule cellule ?

1. Sélectionner la cellule



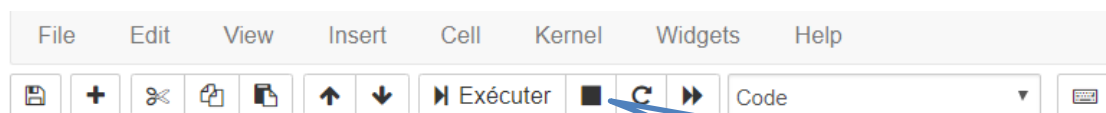
Comment effacer les sorties de toutes les cellules afin de « réinitialiser » le fichier Notebook ?

Première méthode :

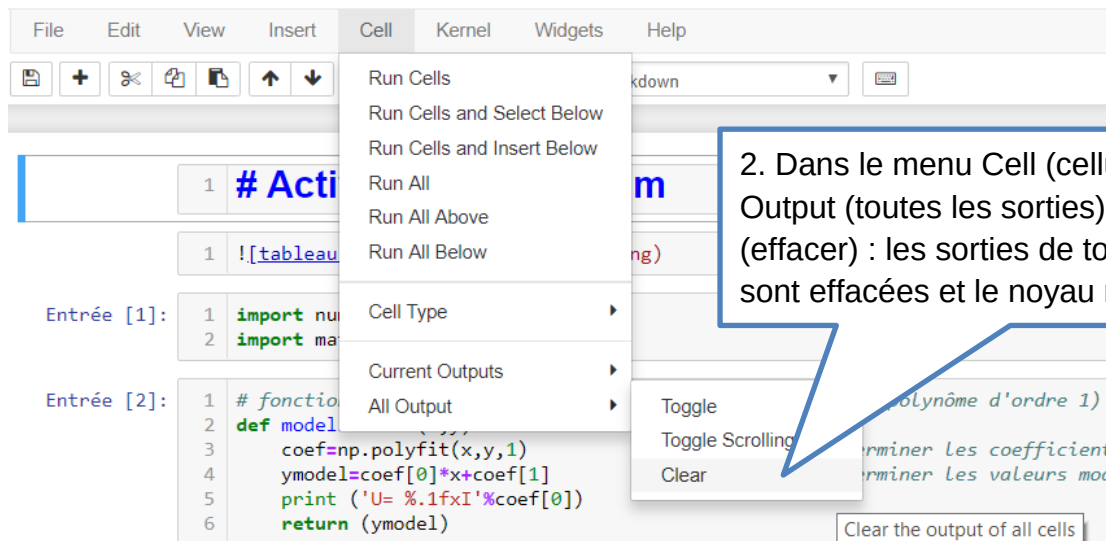


Dans le menu Kernel, cliquer sur Restart & Clear Output (Redémarrer et effacer les sorties) : les sorties de toutes les cellules sont effacées et le noyau est redémarré

Deuxième méthode



1. Interrompre le noyau



2. Dans le menu Cell (cellule), cliquer sur All Output (toutes les sorties) puis sur Clear (effacer) : les sorties de toutes les cellules sont effacées et le noyau reste interrompu.